The Consultative Committee for Space Data Systems

**Draft Recommendation for**
**Space Data System Practices**

# SPACECRAFT ONBOARD INTERFACE SERVICES—DEVICE DATA POOLING SERVICE

## DRAFT RECOMMENDED PRACTICE

## CCSDS 871.1-R-1

## RED BOOK

**June 2007**

# AUTHORITY

| | |
|---|---|
| Issue: | Red Book, Issue 1 |
| Date: | June 2007 |
| Location: | Not Applicable |

**(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)**

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Office of Space Communication (Code M-3)
National Aeronautics and Space Administration
Washington, DC 20546, USA

# STATEMENT OF INTENT

**(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)**

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not in themselves considered binding on any Agency.

CCSDS Recommendations take two forms: **Recommended Standards** that are prescriptive and are the formal vehicles by which CCSDS Agencies create the standards that specify how elements of their space mission support infrastructure shall operate and interoperate with others; and **Recommended Practices** that are more descriptive in nature and are intended to provide general guidance about how to approach a particular problem associated with space mission support. This **Recommended Practice** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommended Practice** is entirely voluntary and does not imply a commitment by any Agency or organization to implement its recommendations in a prescriptive sense.

No later than five years from its date of issuance, this **Recommended Practice** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Practice** is issued, existing CCSDS-related member Practices and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such Practices or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new Practices and implementations towards the later version of the Recommended Practice.

# FOREWORD

**(WHEN THIS RECOMMENDED PRACTICE IS FINALIZED, IT WILL CONTAIN THE FOLLOWING FOREWORD:)**

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Practice is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Organization (NSPO)/Taiwan.
- Naval Center for Space Technology (NCST)/USA.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

# PREFACE

This document is a draft CCSDS Recommended Practice. Its draft status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

# DOCUMENT CONTROL

| Document | Title | Date | Status |
|---|---|---|---|
| CCSDS 871.1-R-1 | Spacecraft Onboard Interface Services—Device Data Pooling Service, Draft Recommended Practice, Issue 1 | June 2007 | Current draft |

# CONTENTS

# 1 INTRODUCTION

## 1.1 PURPOSE AND SCOPE

This document defines the SOIS Device Data Pooling Service. The Device Data Pooling Service provides a standard interface that enables onboard software (applications and high-level services) to access pooled data from simple onboard hardware devices such as sensors and actuators, without explicitly requesting an immediate read of the real device. This avoids multiple reads of the same device when many users require access to the same data. This provides a very basic device read capability that can be used directly by software applications, or can be used as the basis for more capable services, such as those that perform engineering unit conversions on raw data, or monitoring services.

## 1.2 DOCUMENT STRUCTURE

This document comprises three sections:

- section 1, this section, defines common terms used within this document and lists reference documents;
- section 2 (informative) describes the Device Data Pooling Service concept;
- section 3 (normative) defines the Device Data Pooling Service, in terms of the service provided, services expected from underlying layers, and the service interface.

In addition, annex A contains informative references.

## 1.3 DEFINITIONS

### 1.3.1 DEFINITIONS FROM THE OSI REFERENCE MODEL

The Device Data Pooling Service is defined using the style established by the Open Systems Interconnection (OSI) Basic Reference Model (reference [1]). This model provides a common framework for the development of standards in the field of systems interconnection.

The following terms used in this Recommended Practice are adapted from definitions given in reference [1]:

**layer**: a subdivision of the architecture, constituted by subsystems of the same rank.

**service**: a capability of a layer, and the layers beneath it (a service provider), which is provided to the service users at the boundary between the service providers and the service users.

### 1.3.2 TERMS DEFINED IN THIS RECOMMENDED PRACTICE

For the purposes of this Recommended Practice, the following definitions also apply.

**Acquisition:** The act of acquiring a sample for an acquisition order.

**Acquisition Order**: The application-defined order to the service to acquire samples periodically and cache a history of them, so that the application can, on demand, read the samples from the service without having to access the devices directly.

**Application**: Any component of the onboard software that makes use of this service. This includes flight software applications and higher-layer services.

**Data Pool:** A time-ordered cache of samples acquired for an acquisition order. This is similar in concept to a database of the latest available data, or a bulletin board.

**Device**: A real hardware component of the spacecraft such as a sensor or actuator, or a single register within such a component.

**Octet**: An eight-bit word.

**Sample:** A set of values read from different devices at the same time, in response to an acquisition order.

**Value**: A formatted atomic unit of data that is read from or written to a device.

## 1.4 DOCUMENT NOMENCLATURE

The following conventions apply throughout this Recommended Practice:

a) The words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) The word ‘should’ implies an optional, but desirable, specification;

c) The word ‘may’ implies an optional specification;

d) The words ‘is’, ‘are’, and ‘will’ imply statements of fact.

## 1.5 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Practice. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Practice are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Documents.

[1] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard, ISO/IEC 7498-1:1994. 2nd ed. Geneva: ISO, 1994.

[2] *Spacecraft Onboard Interface Services—Device Access Service*. Draft Recommendation for Space Data System Practices, CCSDS 871.0-R-1. Red Book. Issue 1. Washington, D.C.: CCSDS, June 2007.

NOTE – Informative references are contained in annex A.

# 2 SERVICE CONCEPT

## 2.1 OVERVIEW

The SOIS Device Data Pooling Service is defined within the context of the overall SOIS architecture (reference **Error! Reference source not found.**) as one of the services of the Application Support Layer, as illustrated in the following figure.

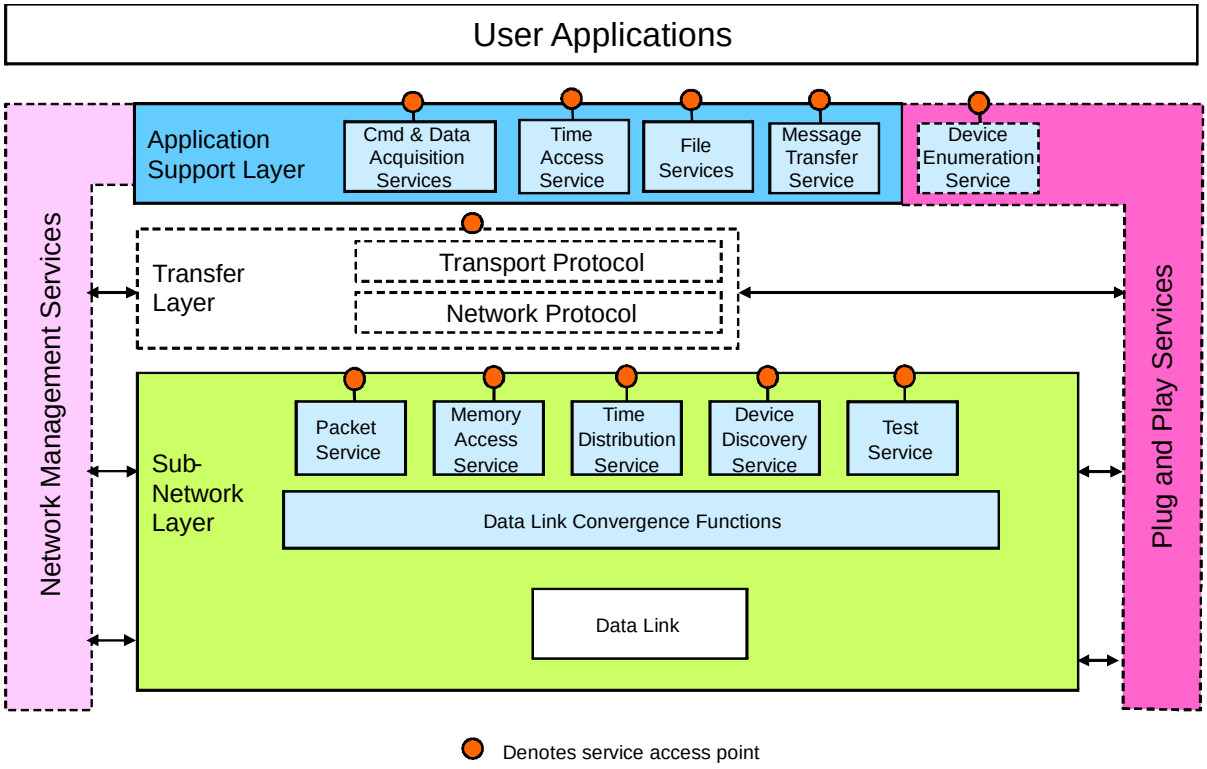


**Figure 2-1: Device Data Pooling Service Context**

NOTE – The SOIS Device Data Pooling Service is one of the services of the Application Support Layer of the SOIS Architecture.

The SOIS Device Data Pooling Service is one of the Command and Data Acquisition services, whose relationships are illustrated in the following figure.

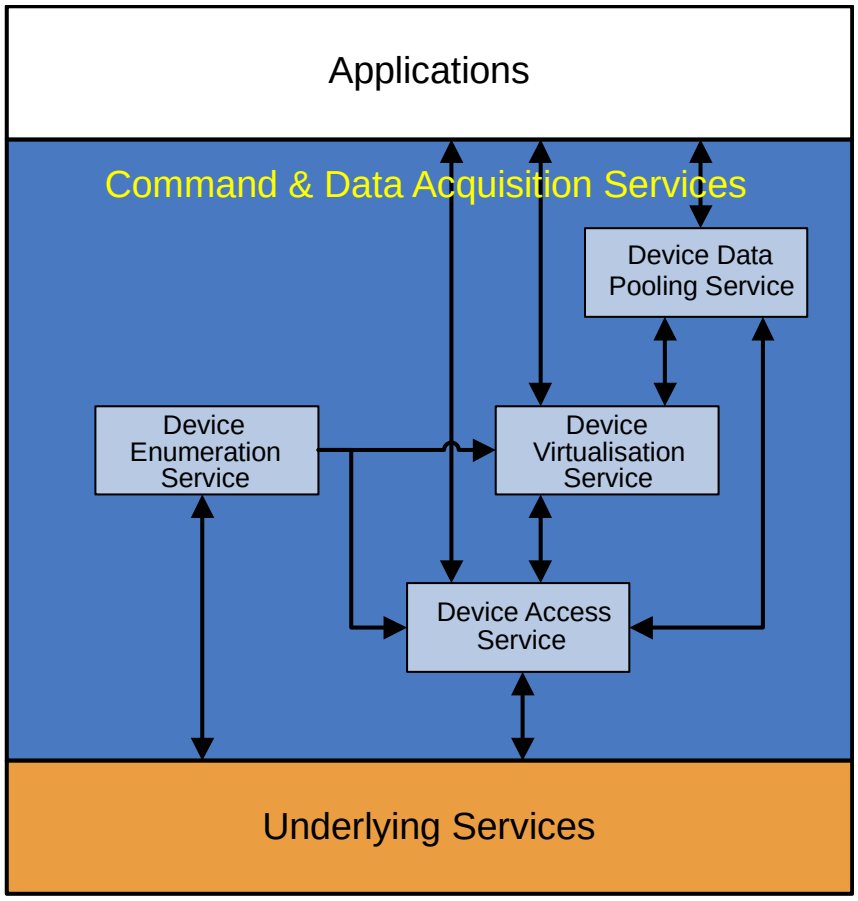


**Figure 2-2: Relationship between SOIS Command and Data Acquisition Services**

NOTE – The SOIS Device Data Pooling Service is one of the Command and Data Acquisition services. It makes use of the SOIS Device Access Service to read values from Devices.

The SOIS Device Data Pooling Service provides a standard interface that enables onboard software (applications and high-level services) to access pooled data from simple onboard hardware devices such as sensors and actuators, without explicitly requesting a read of the real device.

The basic concept underlying the service is that of a *data pool* (see definition in section 1.3.2). A data pool is a periodically acquired cache of samples of values read from a number of devices on the spacecraft. Applications should be able to access samples from that pool, i.e., independently of the precise physical locations of the devices, without requiring detailed knowledge of the electrical interfaces to the devices, and without triggering immediate reads of the devices. This makes it easier to develop the onboard software, enables configuration changes in the spacecraft design to be easily tolerated, and increases the re-use potential of the software.

The attributes of the pool are samples of values from real devices. Each *sample* (see definition in section 1.3.2) is a set of values from a related set of devices, and its acquisition

time. The acquisition interval for a sample is independent from all other samples in the pool. A short, application-defined ordered history of samples is held in the pool.

The relationship between the data pool, the samples, and the individual values read from devices is illustrated in the following figure.

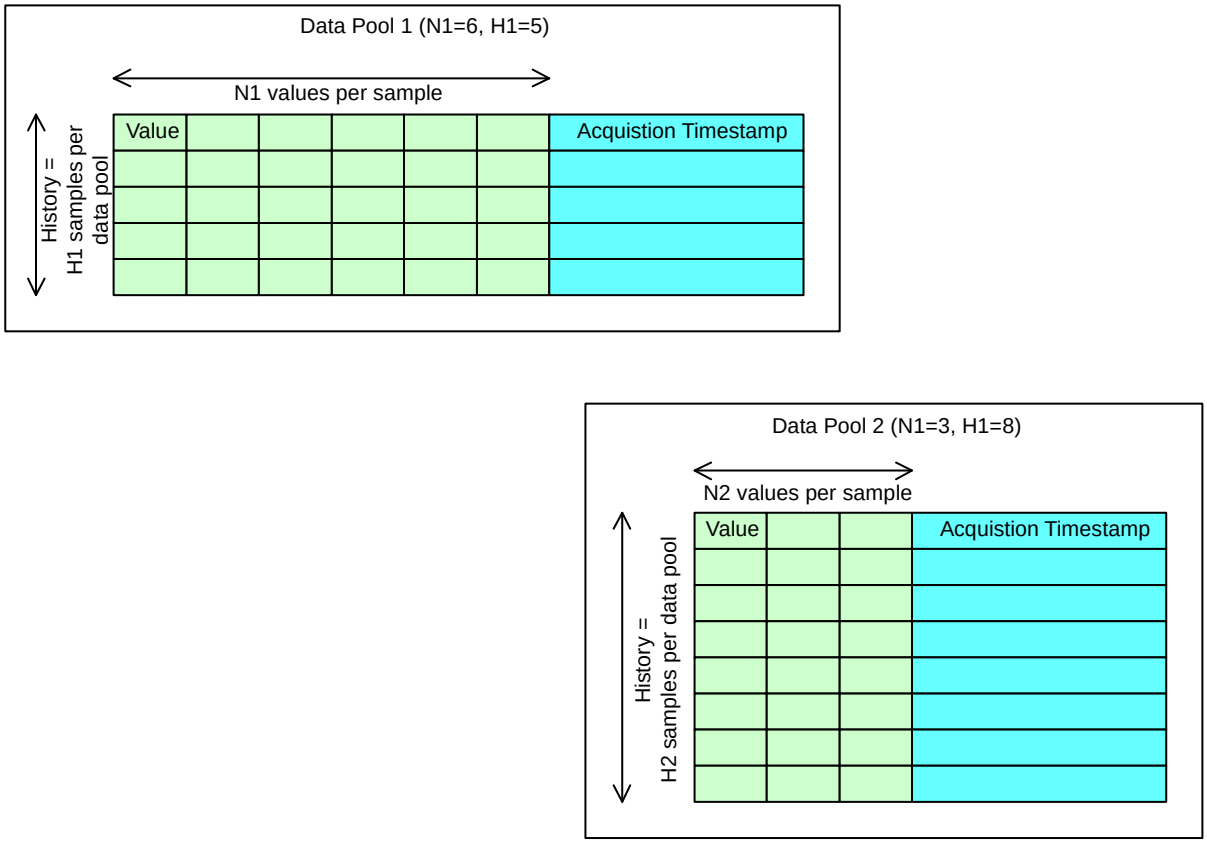


**Figure 2-3: Data Pool Examples**

NOTE – There are N values per sample and H samples in a Data Pool. These vary per Data Pool and are defined when the Acquisition Order is added by the User to the SOIS Device Data Pooling Service.

The acquisitions are initiated on a demand-driven basis. Each user will make an *acquisition order*, which orders the sampling of values from a specified set of devices at a specified acquisition interval, with a specified history of samples maintained.

Multiple acquisitions may be requested by different users on the same devices. The service implementation is responsible for optimising the actual acquisitions performed to avoid over-sampling while still meeting all the users' requirements.

To add an acquisition order for a set of devices, an application provides a set of logical identifiers for the devices, an acquisition interval, and a history size. The service assigns and returns to the application a logical identifier for that acquisition order in order to allow the

application to manage the acquisition order and obtain the samples. The acquiring of samples can be started (at a defined first acquisition time), stopped, and removed by the application.

Upon making an acquisition, the service sends an indication to notify the application of the outcome of the acquisition; if the acquisition was successful, the notification indicates that a new sample is available for reading. The application may then immediately read the sample, defer the reading of the sample to a later time, or ignore the notification altogether. Therefore, there are three use cases for how the application obtains an acquired sample:

1) aperiodic reading of the history (or subset of the history) of the samples;

2) periodic reading of the history (or subset of the history) of the samples (this may or may not be synchronised with the acquisition period);

3) asynchronous read of the history (or subset of the history) of samples driven by the acquisition indication (this does not require the application to be synchronised with the acquisition period).

To request the history of samples, an application provides to the service the logical identifier for the acquisition order and a history depth. The service then returns an indication to the application that contains a time-ordered list of samples, acquisition times, and results of the acquisitions.

The benefit of the service is that its use avoids multiple reads of the same device when many applications require access to the same data, and thus results in reduced bandwidth from the device and probably reduced CPU load. In addition, the application is no longer concerned with the periodic acquisition process, the details of the location of the devices, their physical interfaces, or how they are accessed. As a result, configuration changes involving a change in the physical location of a device, or changes to its electrical interface, do not require changes to the application software using that device.

Although isolated from the details of device location and interface type, the application must still know the format of data read from the device, and the user remains responsible for correctly interpreting those formats.

## 2.2 USE AND OPERATION OF THE DEVICE DATA POOLING SERVICE

Applications invoke the Device Data Pooling Service to acquire periodically a sample of values from devices and then, as required, read the cached samples, rather than to read or write directly to the hardware itself.

From the user's perspective this service will result in applications that are more portable, that are easier to develop, and that can tolerate changes in the spacecraft hardware configuration. From the spacecraft platform implementers' perspective, this service will make it easier to control the access to shared hardware resources.

The Device Data Pooling Service is operated using service requests and service indications passed between the service user and the service provider.

# 3 SERVICE DEFINITION

## 3.1 PROVIDED SERVICE

The Device Data Pooling Service provides the ability to maintain data pools, that is to say a time-ordered cache of samples of values periodically acquired from devices, up to a user-defined maximum history size. New data pools can be added and old data pools removed through the adding and removing of acquisition orders. Periodic acquisitions for the data pools can be started and stopped. The period of the acquisition and the maximum history size are specified when the acquisition order is added. A user-specified history of samples can be read from the service.

## 3.2 EXPECTED SERVICE FROM UNDERLYING LAYERS

The minimum expected service from the underlying layers is the ability to acquire a value from a device based on its device identifier alone. The mechanism by which the value is acquired should be transparent to the Device Data Pooling Service. It is expected that this service is provided by an implementation of the SOIS Device Access Service (reference [2]). The set of valid device identifiers is assumed to be managed by the Device Access Service's Device Identifier Resolution Table managed parameter.

## 3.3 SERVICE INTERFACE

### 3.3.1 GENERAL

The Device Data Pooling Service interface comprises the following primitives:

- DDPS_ADD_ACQUISITION_ORDER.request;
- DDPS_ADD_ACQUISITION_ORDER.indication;
- DDPS_REMOVE_ACQUISITION_ORDER.request;
- DDPS_REMOVE_ACQUISITION_ORDER.indication;
- DDPS_START_ACQUISITIONS.request;
- DDPS_START_ACQUISITIONS.indication;
- DDPS_STOP_ACQUISITIONS.request;
- DDPS_STOP_ACQUISITIONS.indication;
- DDPS_READ_SAMPLES.request;
- DDPS_READ_SAMPLES.indication.

These primitives and their associated parameters are described in the following subsections.

### 3.3.2 DDPS_ADD_ACQUISITION_ORDER.REQUEST

The DDPS_ADD_ACQUISITION_ORDER.request shall be issued by the Device Data Pooling Service user in order to request that an order for the periodic acquisition of samples of values from a number of devices be added (but not enabled), resulting in the creation of a data pool. Multiple periodic acquisitions from the same device may exist at the same time (in different acquisition orders). This primitive contains the set of device identifiers, the acquisition interval, and the size of historical values to retain.

The parameters associated with this primitive are:

– Set of Device_IDs;

– Acquisition_interval;

– History_size.

**Device_ID** identifies a device from which a value is to be read.

**Acquisition_interval** is the time interval between successive acquisitions.

**History_size** is the number of samples that are held in the data pool.

### 3.3.3 DDPS_ADD_ACQUISITION_ORDER.INDICATION

The DDPS_ADD_ACQUISITION_ORDER.indication shall be issued by the Device Data Pooling Service in response to a DDPS_ADD_ACQUISITION_ORDER.request. This primitive contains the assigned acquisition order identifier and indicates whether the request was executed successfully or not.

The parameters associated with this primitive are:

– Result;

– Acquisition_order_ID.

**Result** indicates whether the add request was executed successfully or not. A *No_Error* result implies that the acquisition was successfully added and the associated Acquisition_order_ID parameter is valid. Other results indicate failure conditions such as a Device_ID resolution failure or the history size exceeds some implementation-defined limit.

**Acquisition_order_ID** identifies the added acquisition order.

NOTE – The implementer of the service must provide a mechanism to associate each indication primitive with the request primitive that caused it to be issued. This mechanism may be either explicit or implicit.

### 3.3.4 DDPS_REMOVE_ACQUISITION_ORDER.REQUEST

The DDPS_REMOVE_ACQUISITION_ORDER.request shall be issued by the Device Data Pooling Service user in order to request the removal on an existing acquisition order. An acquisition that has been started and not yet been stopped might not be removed. This primitive contains the assigned acquisition order identifier.

The parameter associated with this primitive is:

– Acquisition_order_ID.

**Acquisition_order_ID** identifies the acquisition order to be removed.

### 3.3.5 DDPS_REMOVE_ACQUISITION_ORDER.INDICATION

The DDPS_REMOVE_ACQUISITION_ORDER.indication shall be issued by the Device Data Pooling Service in response to a DDPS_REMOVE_ACQUISITION_ORDER.request. This primitive indicates whether the request was executed successfully or not.

The parameter associated with this primitive is:

– Result.

**Result** indicates whether the remove request was executed successfully or not. A *No_Error* result implies that the acquisition order was successfully removed. Other results indicate failure conditions such as an Acquisition_order_ID resolution failure or the condition that the acquisition has not yet been completed.

NOTE – The implementer of the service must provide a mechanism to associate each indication primitive with the request primitive that caused it to be issued. This mechanism may be either explicit or implicit.

### 3.3.6 DDPS_START_ACQUISITIONS.REQUEST

The DDPS_START_ACQUISITIONS.request shall be issued by the Device Data Pooling Service user in order to request that a periodic acquisition of samples for an acquisition order be started at a defined onboard time. This primitive contains the assigned acquisition order identifier and the first acquisition time.

The parameters associated with this primitive are:

– Acquisition_order_ID;

– First_acquisition_time.

**Acquisition_order_ID** identifies the acquisition order to be started.

**First_acquisition_time** is the time at which the user wishes the first acquisition to take place.

### 3.3.7 DDPS_START_ACQUISITIONS.INDICATION

The DDPS_START_ACQUISITIONS.indication shall be issued by the Device Data Pooling Service in response to a DDPS_START_ACQUISITIONS.request. This primitive indicates whether the request was executed successfully or not.

The parameter associated with this primitive is:

– Result.

**Result** indicates whether the start request was executed successfully or not. A *No_Error* result implies that the acquisitions were successfully started. Other results indicate failure conditions such as an Acquisition_order_ID resolution failure or that the acquisitions are already started.

NOTE – The implementer of the service must provide a mechanism to associate each indication primitive with the request primitive that caused it to be issued. This mechanism may be either explicit or implicit.

### 3.3.8 DDPS_STOP_ACQUISITIONS.REQUEST

The DDPS_STOP_ACQUISITIONS.request shall be issued by the Device Data Pooling Service user in order to request that a periodic acquisition of samples for an acquisition order be stopped immediately. This primitive contains the assigned acquisition order identifier.

The parameter associated with this primitive is:

– Acquisition_order_ID.

**Acquisition_order_ID** identifies the acquisition order to be stopped.

### 3.3.9 DDPS_STOP_ACQUISITIONS.INDICATION

The DDPS_STOP_ACQUISITIONS.indication shall be issued by the Device Data Pooling Service in response to a DDPS_STOP_ACQUISITIONS.request. This primitive indicates whether the request was executed successfully or not.

The parameter associated with this primitive is:

– Result.

**Result** indicates whether the stop request was executed successfully or not. A *No_Error* result implies that the acquisitions were successfully stopped. Other results indicate failure conditions such as an Acquisition_order_ID resolution failure or that the acquisitions have not been started.

NOTE – The implementer of the service must provide a mechanism to associate each indication primitive with the request primitive that caused it to be issued. This mechanism may be either explicit or implicit.

### 3.3.10 DDPS_READ_SAMPLES.REQUEST

The DDPS_READ_SAMPLES.request shall be issued by the Device Data Pooling Service user in order to request that a user-specified history of cached samples be read, starting with the most recently acquired sample and going back in time from there. This primitive contains the assigned acquisition order identifier and the requested history size.

The parameters associated with this primitive are:

– Acquisition_order_ID;

– History_size.

**Acquisition_order_ID** identifies the acquisition order whose data pool is to be read from.

**History_size** is the number of samples that are to be read from the data pool, starting with the most recently acquired sample and going back in time from there.

### 3.3.11 DDPS_READ_SAMPLES.INDICATION

The DDPS_READ_SAMPLES.indication shall be issued by the Device Data Pooling Service in response to a DDPS_READ_SAMPLES.request. This primitive indicates whether the request was executed successfully or not.

The parameter associated with this primitive are:

– Result;

– Samples.

**Result** indicates whether the read samples request was executed successfully or not. A *No_Error* result implies that the history of samples was successfully read and that the Samples parameter is valid. Other results indicate failure conditions such as an Acquisition_order_ID resolution failure or that the acquisitions have not been started.

**Samples** is a time-ordered set of samples and associated timestamp and outcome of each acquisition.

NOTE – The implementer of the service must provide a mechanism to associate each indication primitive with the request primitive that caused it to be issued. This mechanism may be either explicit or implicit.

# 4 MANAGEMENT INFORMATION BASE

There is no Management Information Base associated with this service.

# 5 SERVICE CONFORMANCE STATEMENT PROFORMA

It is mandatory that, for any implementation claiming to provide this service, this proforma be completed giving details of the capabilities of the implementation.

**Service Conformance Statement**

**SOIS Device Data Pooling Service**

**Implementation Information**

| | |
|---|---|
| Implementer Identification | |
| Implementation Identification | |
| Version | |
| Underlying Data link | |
| Protocol Specification Reference | |
| MIB Reference | |

**Mandatory Features**

| | |
|---|---|
| DDPS_ADD_ACQUISITION_ORDER.request | √ |
| DDPS_ADD_ACQUISITION_ORDER.indication | √ |
| DDPS_REMOVE_ACQUISITION_ORDER.request | √ |
| DDPS_REMOVE_ACQUISITION_ORDER.indication | √ |
| DDPS_START_ACQUISITIONS.request | √ |
| DDPS_START_ACQUISITIONS.indication | √ |
| DDPS_STOP_ACQUISITIONS.request | √ |
| DDPS_STOP_ACQUISITIONS.indication | √ |
| DDPS_READ_SAMPLES.request | √ |
| DDPS_READ_SAMPLES.indication | √ |

**Optional Features**

| | |
|---|---|
| None | |

**Other Information**

| | |
|---|---|
| N/A | |

# ANNEX A

# INFORMATIVE REFERENCES

[A1] *Spacecraft Onboard Interface Services*. Report Concerning Space Data System Standards, CCSDS 850.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, June 2007.

NOTE – Normative references are listed in 1.4.